



# Contacts

**Link submit:** <https://www.hackerrank.com/challenges/contacts/problem>

**Solution:**

C++	<a href="https://ideone.com/wrKkQ2">https://ideone.com/wrKkQ2</a>
Java	<a href="https://ideone.com/J7FHUx">https://ideone.com/J7FHUx</a>
Python	<a href="https://ideone.com/aZvYBm">https://ideone.com/aZvYBm</a>

**Tóm tắt đề:**

Bạn sẽ tạo ra một ứng dụng quản lý thông tin liên lạc. Ứng dụng này phải cung cấp 2 thao tác:

- add name: với name là tên của một liên hệ. Thao tác này lưu lại thông tin liên lạc của người tên name như một liên hệ mới.
- find partial, với partial là một chuỗi miêu tả một phần của các thông tin cần tìm kiếm. Thao tác này trả về số người trong danh sách liên lạc có tên có phần đầu là partial.

**Input**

Dòng đầu chứa số nguyên dương  $n$  - số thao tác cần thực hiện.

Mỗi dòng trong  $n$  dòng tiếp theo sẽ chứa miêu tả thao tác cần thực hiện.

**Ràng buộc:**

- $1 \leq n \leq 10^5$
- $1 \leq \text{độ dài name} \leq 21$
- $1 \leq \text{độ dài partial} \leq 21$
- name và partial được đảm bảo chỉ chứa các ký tự thường tiếng Anh.
- Không có name nào xuất hiện trên 1 lần.

**Output**

Với mỗi thao tác find partial, in ra trên một dòng là kết quả trả về của thao tác.

**Ví dụ:**

4	2
add hack	0
add hackerrank	
find hac	
find hak	

**Giải thích:**

Chuỗi hack và hackerrank đều có hac ở đầu nhưng không có hak.

**Hướng dẫn giải:**

Sử dụng cây trie để lưu tên liên lạc.

Mỗi nút trên cây ngoài chứa liên kết tới 26 nút con thì còn lưu thêm số lượng name đi qua nút đó trong quá trình add, đây chính là số name có phần tiền tố là đường đi từ gốc tới nút đó.

Thao tác truy vấn: duyệt cây trie theo chuỗi partial, nếu không đi hết được chuỗi thì trả về 0, ngược lại trả về giá trị count của nút duyệt cuối cùng.

**Độ phức tạp:**  $O(\text{string\_length} * N)$  với string\_length là độ dài lớn nhất của chuỗi và N là số lượng truy vấn.